CLAIMS

What is claimed is:

1. A method for object-oriented management of serializable objects, the method comprising:

identifying an object, wherein the object comprises a set of attributes;

associating a class version identifier with the object, wherein the class version identifier identifies the object as an instance of a specific version of a class; and

associating an attribute version identifier with an attribute in the set of attributes.

2. The method of claim 1 wherein an attribute version identifier represents a version of a class for an associated attribute in which the associated attribute was initially declared within the class.

3. The method of claim 1 further comprising:

writing a data stream representing an object serialization of the object, wherein the data stream comprises the class version identifier of the object, an attribute value for an attribute in the set of attributes, and an attribute version identifier for an attribute in the set of attributes.

4. The method of claim 3 further comprising:

writing a class identifier for the class of the object into the data stream.

5. The method of claim 3 further comprising:

writing an attribute count indicating a number of attributes from the set of attributes that were written into the data stream.

6. The method of claim 1 further comprising:

reading a data stream representing a serialized object, wherein the data stream comprises a serialized class version identifier, a set of serialized attribute values, and a set of serialized attribute version identifiers, wherein serialized attribute version identifiers in the set of serialized attribute version identifiers are paired with serialized attribute values in the set of serialized attribute values.

7. The method of claim 6 further comprising:

reading a class identifier for the serialized object from the data stream; and

instantiating the object in accordance with the class identifier, wherein the class version identifier of the object and the serialized class version identifier of the serialized object may differ.

8. The method of claim 6 further comprising:

reading an attribute count for the set of serialized attribute values from the data stream.

9. The method of claim 7 further comprising:

mapping attributes between the object and the serialized object; and

storing serialized attribute values from the data stream in the object.

10. The method of claim 9 further comprising:

in response to a determination that a serialized attribute version identifier is greater than or subsequent to the class version identifier of the object, refraining from storing in the object a serialized attribute value associated with the serialized attribute version identifier.

11. The method of claim 9 further comprising:

in response to a determination that the class version identifier of the object is greater than or subsequent to the serialized class version identifier, storing default attribute values in the object for attributes in the object that are associated with an attribute version identifier that is greater than or subsequent to the serialized class version identifier.

12. A method for providing backwards and forwards compatibility between different versions of serialized object data, the method comprising:

identifying an object, wherein the object comprises a set of attributes, wherein each attribute in the set of attributes is associated with a version identifier, and wherein the object is an instance of a first version of a class;

writing a data stream representing serialization of the object's attributes and associated version identifiers;

reading a data stream representing a serialized object into a new object instance of a second version of a class; and

refraining from storing attributes from the data stream into the new object instance that are not represented in the new object instance while reading the data stream.

13. The method of claim 12 further comprising:

specifying default values for attributes in the new object instance for which attribute values were not read from the data stream.

14. A computer program product on a computer readable medium for use in a data processing system for object-oriented management of serializable objects, the computer program product comprising:

instructions for identifying an object, wherein the object comprises a set of attributes;

instructions for associating a class version identifier with the object, wherein the class version identifier identifies the object as an instance of a specific version of a class; and

instructions for associating an attribute version identifier with an attribute in the set of attributes.

15. The computer program product of claim 14 wherein an attribute version identifier represents a version of a class for an associated attribute in which the associated attribute was initially declared within the class.

16. The computer program product of claim 14 further comprising:

instructions for writing a data stream representing an object serialization of the object, wherein the data stream comprises the class version identifier of the object, an attribute value for an attribute in the set of attributes, and an attribute version identifier for an attribute in the set of attributes.

17. The computer program product of claim 16 further comprising:

instructions for writing a class identifier for the class of the object into the data stream.

18. The computer program product of claim 16 further comprising:

instructions for writing an attribute count indicating a number of attributes from the set of attributes that were written into the data stream.

19. The computer program product of claim 14 further comprising:

instructions for reading a data stream representing a serialized object, wherein the data stream comprises a serialized class version identifier, a set of serialized attribute values, and a set of serialized attribute version identifiers, wherein serialized attribute version identifiers in the set of serialized attribute version identifiers are paired with serialized attribute values in the set of serialized attribute values.

20. The computer program product of claim 19 further comprising:

instructions for reading a class identifier for the serialized object from the data stream; and

instructions for instantiating the object in accordance with the class identifier, wherein the class version identifier of the object and the serialized class version identifier of the serialized object may differ.

21. The computer program product of claim 19 further comprising:

instructions for reading an attribute count for the set of serialized attribute values from the data stream.

22. The computer program product of claim 20 further comprising:

instructions for mapping attributes between the object and the serialized object; and

instructions for storing serialized attribute values from the data stream in the object.

23. The computer program product of claim 22 further comprising:

instructions for refraining from storing in the object a serialized attribute value associated with the serialized attribute version identifier in response to a determination that a serialized attribute version identifier is greater than or subsequent to the class version identifier of the object.

24. The computer program product of claim 22 further comprising:

instructions for storing default attribute values in the object for attributes in the object that are associated with an attribute version identifier that is greater than or subsequent to the serialized class version identifier in response to a determination that the class version identifier of the object is greater than or subsequent to the serialized class version identifier.

25. A computer program product on a computer readable medium for use in a data processing system for providing backwards and forwards compatibility between different versions of serialized object data, the computer program product comprising:

instructions for identifying an object, wherein the object comprises a set of attributes, wherein each attribute in the set of attributes is associated with a version identifier, and wherein the object is an instance of a first version of a class;

instructions for writing a data stream representing serialization of the object's attributes and associated version identifiers;

instructions for reading a data stream representing a serialized object into a new object instance of a second version of a class; and

instructions for refraining from storing attributes from the data stream into the new object instance that are not represented in the new object instance while reading the data stream.

26. The computer program product of claim 25 further comprising:

instructions for specifying default values for attributes in the new object instance for which attribute values were not read from the data stream.

27. An apparatus for object-oriented management of serializable objects, the apparatus comprising:

means for identifying an object, wherein the object comprises a set of attributes;

means for associating a class version identifier with the object, wherein the class version identifier identifies the object as an instance of a specific version of a class; and

means for associating an attribute version identifier with an attribute in the set of attributes.

28. The apparatus of claim 27 further comprising:

means for writing a data stream representing an object serialization of the object, wherein the data stream comprises the class version identifier of the object, an attribute value for an attribute in the set of attributes, and an attribute version identifier for an attribute in the set of attributes.

29. The apparatus of claim 27 further comprising:

means for reading a data stream representing a serialized object, wherein the data stream comprises a serialized class version identifier, a set of serialized attribute values, and a set of serialized attribute version identifiers, wherein serialized attribute version identifiers in the set of serialized attribute version identifiers are paired with serialized attribute values in the set of serialized attribute values.

30. An apparatus for providing backwards and forwards compatibility between different versions of serialized object data, the apparatus comprising:

means for identifying an object, wherein the object comprises a set of attributes, wherein each attribute in the set of attributes is associated with a version identifier, and wherein the object is an instance of a first version of a class;

means for writing a data stream representing serialization of the object's attributes and associated version identifiers;

means for reading a data stream representing a serialized object into a new object instance of a second version of a class; and

means for refraining from storing attributes from the data stream into the new object instance that are not represented in the new object instance while reading the data stream.